

Entrega Tema 2 – Base de Datos

Diseño Lógico de Base de Datos



“Federación de Parchís”

Desarrollo de Aplicaciones Web (DAW)

Base de Datos

Cesur

*Curso 1º
2023-2024*

Andy López Rey

10/03/2024

Modelo Entidad - Relación

Extendido y

Modelo Relacional

Generalización / Especialización

- Especialización. Superclase Personas se generan varias entidades hijas (Jugador, Árbitro).
- Disjunta/exclusiva. No se permite que un jugador participe como árbitro y jugador a la vez
- Total. La entidad padre se ha de materializar en una de las entidades hijas sí o sí

Resultado: Especialización disjuntiva exclusiva total.

Modelo Entidad – Relación Extendido

Se explicarán brevemente cada relación entre las entidades y cardinalidades.

- Persona (entidad padre) → Federación. Relación 1:N y 1:1. Sabemos que cada jugador y/o árbitro sólo puede estar asociado a una y sólo una federación porque el enunciado ya nos lo dice. Y una federación puede tener 1 o N cantidad de jugadores/árbitros. No se ha diseñado como una relación 0:N porque una federación sin jugadores o árbitros no estaría ni siquiera en un torneo de parchís. No interesa esta situación. Tendremos una cardinalidad máxima de 1:N.
- Persona → Hotel. Relación 0:M y 0:N. Tenemos una relación 0:N/M en ambos sentidos. Las personas pueden quedarse en un hotel o no, dentro de un abanico de opciones que es proporcionado por la asociación de parchís. Y los hoteles igual por el mismo razonamiento. Tenemos una cardinalidad máxima N:M.
- Jugador → Partida. Relación N:M y 1:N. Sabemos que en una partida hay 4 jugadores como máximo. Además, para que una partida se de y haya un ganador como tal, ha de ser entre mínimo 2 personas. Por esto tendremos una relación N:M. Por otro lado, el enunciado nos dice que un jugador ha de haber jugado al menos 1 partida para considerarse como tal. Y cada jugador puede haber jugado 1 o muchas partidas; por tanto una 1:N. Cardinalidad máxima N:M.
- Jugador → Movimiento. Relación 1:1 y N:M. Si queremos relacionar los movimientos que cada jugador hace lo tendremos que hacer a través de esta relación. Los movimientos se almacenarán para cada jugador (1:1) y cada jugador ha de tener más de un movimiento como mínimo. No se da la posibilidad de 1 movimiento como mínimo porque se asume que el jugador ha completado la partida o al menos la ha comenzado hasta cierto punto.
- Árbitro → Partida. Tenemos una relación 1:1 y 1:N. Ambas partes de la relación nos la da ya el enunciado. Sabemos que una partida sólo podrá ser arbitrada por 1 árbitro. Y cada árbitro ha de arbitrar como mínimo 1 partida. Misma situación que con Jugador. Cardinalidad máxima 1:N.
- Partida → Movimiento. Relación 1:1 y N:M con entidad débil Movimiento. Los movimientos de cada partida son almacenados individualmente para esa partida sin relación a otras (1:1). Por otro lado, cada partida ha de consistir de varios movimientos para poder acabar. No podemos tener una partida que finalice con 1 movimiento. De aquí la relación N:M. Cardinalidad máxima N:M.
- Partida → Sala. Relación 0:M y 0:N. Se pueden celebrar ninguna o muchas partidas en las salas. Un torneo de parchís puede tener una gran cantidad de participantes y audiencia. Es necesario realizar varias partidas simultáneamente en las salas. Por la misma razón habrá 0 o ninguna salas en las que se celebrarán partidas. Cardinalidad máxima: N:M.
- Hotel → Sala. Relación 1:M y 0:N. Las salas pueden ser reservadas por 1 hotel como mínimo (para poder existir) pero pueden haber muchos hoteles que reservan salas para el torneo. Y los hoteles pueden reservar 0 salas si el torneo decide no celebrarse en el hotel de ese jugador/árbitro o varias salas. Los hoteles no suelen tener una única sala de conferencia.

Transformación del diagrama al Modelo Relacional

Claves primarias / Primary Keys: subrayado.

Claves foráneas / Foreign Keys: subrayadas y con asterisco (*).

Resultado final después de realizar la [Transformación](#) al [Modelo Entidad – Relación Extendido](#).

Personas (DNI, Nombre, Apellido_1, Apellido_2, Fecha_Nacimiento, ID_Federación*)

Teléfonos_Personas (DNI*, Teléfonos)

Jugadores (ID_Federado, DNI_Jugador*, Licencia_c_Árb, Año_Federado, Pos_Rank_Nac)

Árbitros (ID_Colegiado, DNI_Árbitro*, Núm_Años_Árb)

Federaciones (ID_Federación, Nombre_Región, Fecha_Fundación, Núm_Total_Jugadores, Núm_Total_Árbitros)

Hoteles (ID_Hotel, Nombre_Hotel, Categoría, Dirección, Núm_Total_Habitaciones)

Teléfonos_Hotel (ID_Hotel*, Teléfonos)

Hospedan (DNI*, ID_Hotel*, Núm_Habitación, Fecha_Entrada_Habitación, Fecha_Salida, Núm_Hab)

Partidas (ID_Partida, Fecha_Hora, Duración, ID_Colegiado*)

Jugar_Partidas (ID_Federado*, ID_Partida*, Color_Fichas, Puesto_Jugador)

Movimientos (Núm_Orden_Jugada, ID_Federado*, ID_Partida*, Núm_Casilla_Inicial, Núm_Casilla_Final, Comentario_Movimiento)

Salas (Núm_Sala, ID_Hotel*, Capacidad_Sala, Descripción)

Partidas_en_Salas (ID_Partida*, Núm_Sala*)

Hoteles_Salas (ID_Hotel*, Núm_Sala*)

Aclaraciones y Limitaciones

Jugadores: Los jugadores no pueden participar como árbitros a la vez o viceversa. Sin embargo, hay jugadores que pueden tener la licencia de árbitro. Por este motivo existe el atributo en Jugadores "Licencia_c_Árbitro". Se trata de un atributo de tipo booleano que dirá si tienen (True) o no (False) la licencia de árbitro. Es algo que se tendrá que tener en cuenta a la hora de realizar el modelo físico de la base de datos.

Entidades débiles: Movimiento y Sala. Sala es una entidad débil de tipo existencial. Si no se reservan/hospedan partidas en los hoteles, las salas no existen. La entidad débil Movimiento por otro lado es de tipo identificación. Además de la dependencia existencial sobre Partida, también necesita la entidad fuerte para crear sus propios atributos; jugadores.

Tenemos dos tablas teléfonos resultado del atributo multivalorado Teléfonos en Personas y Hoteles. Se ha especificado sobre qué tabla pertenecen.

Interesa saber qué movimiento hace cada jugador durante la partida. Esto se podrá saber gracias a la relación entre Jugador y Movimiento. Por las reglas de transformación, la entidad débil Movimiento hereda la clave principal de jugador. De esta forma quedan ancladas.

La entidad Hotel tiene como atributo Categoría. Se pide en el modelo que tenga un valor entre 1 y 5. Esto a nivel de diseño lógico con modelo E-R extendido no se puede lograr. Al igual que con el caso del atributo de Licencia_c_Árbitro es una limitación que se deberá de solucionar a la hora de realizar la DDL. Lo mismo ocurre con el atributo Color_Fichas de Jugador → Partida.

En lo referente a este último atributo en el modelo se pide explícitamente que un jugador puede cambiar el color de sus fichas de partida a partida, pero no durante la misma partida. Esto deberá realizarse a nivel de lenguaje de programación.

Normalización (FN1 - FN3)

FN1

En la primera fase normal o FN1 se prohíbe que en una tabla haya atributos que puedan tomar más de un valor a la vez.

Aquellos atributos que puedan tomar más de un valor a la vez, deberán de ser eliminados de la tabla y formarán una entidad completamente nueva por su propia cuenta.

Al aplicarla al modelo construido no se detectaron atributos que pudieran tener múltiples valores.

FN2

Para aplicar la segunda forma normal o FN2 el modelo ha de estar en FN1. Hemos comprobado anteriormente que efectivamente lo está.

La FN2 dicta que cada atributo que no forma parte de la clave tiene que tener una dependencia funcional completa de la clave principal o Primary Key; aquellos atributos que están subrayados.

Una vez se identifican los atributos que no dependen de la clave principal, se formará con ellos una nueva entidad y se eliminarán de la original. Además, la clave primaria de la nueva entidad estará formada por parte de la antigua de la que sí dependían funcionalmente los atributos (en caso de ser compuestas).

Analizaremos relación por relación si está o no en FN2.

- **Personas.** Contiene toda la información común relacionada de Árbitro y Jugador. Todos los atributos que contiene son personales. Por tanto estarán ligados a cada persona; a la clave primaria: DNI.
- **Teléfonos_Personas.** Se trata de una relación originada por la transformación de un atributo multivalorado. Al tener dos atributos claves no es necesario aplicar normalización. Por definición ya lo está.
- **Jugadores.** Se trata de una relación con una clave compuesta: ID_Federado y DNI_Jugador. Todos los atributos que no formen parte de la clave principal han de tener una dependencia funcional completa de la compuesta. Todos los atributos están relacionados con la identidad de la persona, pero también de la identidad de la persona como jugador. No podremos saber su posición en el ranking, por ejemplo, únicamente sabiendo su DNI. También necesitaremos su código de identificación como jugador.
- **Árbitros.** Caso exactamente igual al de Jugadores.
- **Federaciones.** A partir del código de identificación de la federación podremos saber unívocamente la información del resto de los atributos.
- **Hoteles.** Dado un código o ID_Hotel podremos saber el resto de la información.
- **Teléfonos_Hotel.** Mismo caso que Teléfonos_Personas.
- **Hospedan.** Hay un único atributo no clave: Fecha_Salida. Es un atributo que podremos saber únicamente teniendo la información de la clave compuesta.
- **Partidas.** Debido a que las partidas han de estar obligatoriamente arbitradas por un árbitro, podemos sacar los atributos con la ayuda de la clave compuesta. En caso contrario podría darse el caso de que no se cumpla la dependencia funcional completa.
- **Jugar_Partidas.** Es una relación originada por la transformación de cardinalidad máxima N:M entre Jugador → Partida. Como cada jugador debe de tener un color de ficha igual para cada partida (individual) y quedará en un puesto determinada, es necesario no sólo tener el código de identificación de la partida, sino también el de jugador. Cumple la dependencia funcional completa y por ende está en FN2.
- **Movimientos.** La entidad débil movimiento. La información está relacionada con los

movimientos que cada jugador hace. Por ende no podremos saber la información de los movimientos si no tenemos también el código que identifica a ese jugador.

- **Salas.** Los atributos dependerán unívocamente de un código de identificación, de la clave compuesta. Pues hay dependencia existencial de Hotel.
- **Partidas_en_Salas.** No tenemos atributos no clave para analizar. No se normaliza; ya lo está por definición.
- **Hoteles_Salas.** Mismo caso que la relación anterior.

Podemos concluir que el modelo está en FN2.

FN3

La tercera y última fase de normalización o FN3 se da cuando los elementos están en FN2 y además todos los atributos no principales dependen directamente de la clave principal.

Es decir, no deben de haber atributos no principales que dependan de forma transitiva de la clave principal.

De esta forma la FN3 asegura que todos los atributos no clave dependan directamente de la clave y, en ningún caso, dependa de ella de forma indirecta. Por tanto los atributos no claves no pueden pasar por un tercero para así entonces poder depender de la clave primaria.

La definición "formal" sería: $A \rightarrow (\text{determina un único}) B \rightarrow (\text{Determina un único}) C$, pero B (NO determina un único) $\rightarrow A$.

- **Personas** (DNI, Nombre, Apellido_1, Apellido_2, Fecha_Nacimiento, ID_Federación*)

Aclaración: Nombre, Apellido_1 y Apellido_2 son resultado de la transformación de un atributo compuesto. Pero para nuestro análisis tomaremos estos tres como un único atributo: Nombre. En caso contrario se rompería la FN3, pues un único Nombre_Pila no determina un DNI y Apellido; ya que pueden repetirse muchos nombres, apellidos etc. Y se tendría que crear una nueva tabla sólo con los nombres.

$DNI \rightarrow \text{Nombre}$. Veremos si $\text{Nombre} \rightarrow$ algún atributo no clave (dependencia transitiva):

$\text{Nombre} \rightarrow \text{Fecha_Nacimiento}$. Cada persona tendrá una fecha de nacimiento. Y $\text{Nombre} \rightarrow DNI$, pero $\text{Fecha_Nacimiento} \text{ NO } \rightarrow DNI$. No hay forma de determinar unívocamente algún atributo no clave a partir de sólo la fecha de nacimiento.

- **Teléfonos_Personas** (DNI*, Teléfonos). No tenemos atributos no clave para analizar. No se normaliza; ya lo está por definición.
- **Jugadores** (ID_Federado, DNI_Jugador*, Licencia_c_Árb, Año_Federado, Pos_Rank_Nac)

Jugadores posee la clave compuesta ID_Federado y DNI_Jugador. Por tanto tendremos que analizar la situación en base a ambas claves.

$ID_Federado + DNI_Jugador \rightarrow \text{Año_Federado}$. Analicemos la situación.

$\text{Año_Federado} \rightarrow \text{Pos_Rank_Nac}$. Pero a través de $\text{Pos_Rank_Nac} \text{ NO } \rightarrow ID_Federado + DNI_Jugador$.

Pero si analizamos ID_Federado + DNI_Jugador → Licencia_c_Árbitro, vemos que sí hay una dependencia transitiva. El atributo licencia como árbitro no depende de ID_Federado, sino de DNI_Jugador y ID_Colegiado, en todo caso. Por tanto es un atributo que no debería de estar en esta relación.

Gracias a la normalización hemos podido optimizar esta relación. Pero a la vez nos surge un problema. Sabemos que un jugador es posible que sea árbitro; pero no a la vez. Y debemos saberlo de alguna forma de antemano. Es una limitación que tendrá nuestro modelo y se deberá de resolver a nivel de programación.

- **Árbitros** (ID_Colegiado, DNI_Árbitro*, Núm_Años_Árb)

ID_Colegiado + DNI_Árbitro → Núm_Años_Árb. A partir de los atributos claves podremos determinar sin problema el atributo no clave. No hay más atributos a analizar la transitoriedad.

- **Federaciones** (ID_Federación, Nombre_Región, Fecha_Fundación, Núm_Total_Jugadores, Núm_Total_Árbitros)

ID_Federación → Nombre_Región. Analicemos si Nombre_Región → algún atributo no clave:

Nombre_Región → Fecha_Fundación. No hay posibilidad de determinar algún atributo no clave unívocamente a través de Fecha_Fundación.

Nombre_Región → (Núm_Total_Jugadores, Núm_Total_Árbitros). Nuevamente sólo con el número de jugadores o de árbitros no podremos determinar el nombre de la región, por ejemplo. (Núm_Total_Jugadores, Núm_Total_Árbitros) NO → Nombre_Región.

ID_Federación → Fecha_Fundación. A partir de la identificación de la federación podremos saber sin problema en qué año se fundó. Veamos ahora si Fecha_Fundación → algún atributo no clave.

Fecha_Fundación → (Núm_Total_Jugadores, Núm_Total_Árbitros). Pero (Núm_Total_Jugadores, Núm_Total_Árbitros) NO → Fecha_Fundación. Caso similar al de antes. No se dan las condiciones de transitividad.

- **Hoteles** (ID_Hotel, Nombre_Hotel, Categoría, Dirección, Núm_Total_Habitaciones)

ID_Hotel → Nombre_Hotel. Veamos si Nombre_Hotel → algún atributo no clave.

Nombre_Hotel → (Categoría, Dirección, Núm_Total_Habitaciones). No es posible determinar algún atributo no clave unívocamente a partir de ninguno de esos atributos. Debemos de utilizar sí o sí la clave principal para ello. Por tanto podemos decir que no hay transitividad.

- **Teléfonos_Hotel** (ID_Hotel*, Teléfonos). No tenemos atributos no clave para analizar. No se normaliza; ya lo está por definición.
- **Hospedan** (DNI*, ID_Hotel*, Núm_Habitación, Fecha_Entrada_Habitación, Fecha_Salida, Núm_Hab).

A partir de Fecha_Salida o Núm_Hab no es posible determinar unívocamente ni el uno ni el otro. Se ha de usar la clave compuesta.

- **Partidas** (ID_Partida, Fecha_Hora, Duración, ID_Colegiado*).

ID_Partida → Fecha_Hora. Sólo a partir de la clave compuesta podremos determinar la Fecha_Hora. Pero veamos si Fecha_Hora → Duración.

Duración NO → ID_Partida. Por tanto al no haber transitividad podemos decir que se cumple la FN3.

- **Jugar_Partidas** (ID_Federado*, ID_Partida*, Color_Fichas, Puesto_Jugador).

Color_Fichas o Puesto_Jugador NO → ID_Partida. No es posible determinar unívocamente una partida sólo a partir de alguno de esos atributos.

- **Movimientos** (Núm_Orden_Jugada, ID_Federado*, ID_Partida*, Núm_Casilla_Inicial, Núm_Casilla_Final, Comentario_Movimiento).

Tenemos el caso de una clave compuesta: Núm_Orden_Jugada + ID_Federado. Para poder determinar cada atributo se deberá de saber sobre qué jugada se habla pero también de qué jugador. Y esto sólo lo podremos saber a través de la clave compuesta. No es posible determinarlo unívocamente con Núm_Casilla_Inicial NO → Comentario_Movimiento, viceversa, etc.

- **Salas** (Núm_Sala, ID_Hotel*, Capacidad_Sala, Descripción).

Capacidad_Sala o Descripción no podrán determinar unívocamente Descripción o Capacidad_Sala. Necesitan de ID_Hotel.

- **Partidas_en_Salas** (ID_Partida*, Núm_Sala*). No tenemos atributos no clave para analizar. No se normaliza; ya lo está por definición.
- **Hoteles_Salas** (ID_Hotel*, Núm_Sala*). No tenemos atributos no clave para analizar. No se normaliza; ya lo está por definición.

Se concluye que el modelo relacional presentado estará en FN3 si se elimina el atributo “Licencia_c_Árbitro” de la relación Jugador. Así pues, el modelo quedaría:

Personas (DNI, Nombre, Apellido_1, Apellido_2, Fecha_Nacimiento, ID_Federación*)

Teléfonos_Personas (DNI*, Teléfonos)

Jugadores (ID_Federado, DNI_Jugador*, Año_Federado, Pos_Rank_Nac)

Árbitros (ID_Colegiado, DNI_Árbitro*, Núm_Años_Árb)

Federaciones (ID_Federación, Nombre_Región, Fecha_Fundación, Núm_Total_Jugadores, Núm_Total_Árbitros)

Hoteles (ID_Hotel, Nombre_Hotel, Categoría, Dirección, Núm_Total_Habitaciones)

Teléfonos_Hotel (ID_Hotel*, Teléfonos)

Hospedan (DNI*, ID_Hotel*, Núm_Habitación, Fecha_Entrada_Habitación, Fecha_Salida, Núm_Hab)

Partidas (ID_Partida, Fecha_Hora, Duración, ID_Colegiado*)

Jugar_Partidas (ID_Federado*, ID_Partida*, Color_Fichas, Puesto_Jugador)

Movimientos (Núm_Orden_Jugada, ID_Federado*, ID_Partida*, Núm_Casilla_Inicial, Núm_Casilla_Final, Comentario_Movimiento)

Salas (Núm_Sala, ID_Hotel*, Capacidad_Sala, Descripción)

Partidas_en_Salas (ID_Partida*, Núm_Sala*)

Hoteles_Salas (ID_Hotel*, Núm_Sala*)